80. (New claim) The system for processing data according to claim 71 further including means for freeing entries of the processor requests in the return queue, after transmitting corresponding responses from the peripheral devices to the requesting processors.

A1 end

REMARKS

In response to the above identified office action the Applicants have canceled pending claims 1-38 and have added claims 39-80. Claims 39-80 remain in the application. Reconsideration of this application is respectfully requested.

Restriction Requirement

The Examiner has required cancellation of the non-elected claims 27-34. As noted above, the Applicants have canceled all of their pending claims, including the non-elected claims 27-34. The Applicants respectfully request examination of their new claims 39-80, which are believed to be directed to a single invention, namely, a system for processing data and method thereof.

Drawings

After careful review of the submitted drawings, the Applicants have identified instances of missing reference numerals and mislabeled boxes in Figure 10. A marked up copy of Figure 10, which shows the corrections in red, is submitted for

examination purposes. A formal set of drawings will be submitted, once the Examiner allows the outstanding claims.

Objection to the Specification and Rejection of Claims under 35 U.S.C. 112, first paragraph

The Examiner has objected to the specification under 35 U.S.C. 112, first paragraph, for failing to provide an enabling disclosure for claims 16 and 26 and has rejected these claims under the same ground. In view of the cancellation of claims 16 and 26, the Applicants find it unnecessary to address the merits of this rejection. However, the Applicants reserve the right to address the merits at a later time, if necessary.

Rejection of Claims under 35 U.S.C. 102 (e) and 35 U.S.C. 103

The Examiner has rejected claims 1-11, 14-15, 17-20 and 35-38 under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 5,546,546 issued to Bell et al. (The Bell patent).

In view of the Bell patent, the Applicants have canceled their pending claims 1-38 and have submitted new claims 39-80, which in their view more clearly point out the novel features of the present invention. The Applicants' new claims relate to a data processing system and method that increase data transaction throughput between one or more requesting processors and a plurality of peripheral devices, by separately queuing processor requests directed to each one of the peripheral devices in entries of a plurality of pending queues that correspond to each one of the peripheral devices.

More particularly, claims 39-60 are directed to a data processing system, which includes one or more requesting processors that generate processor requests. The processor requests are directed to a plurality of peripheral devices that accept and respond to such processor requests. In an exemplary embodiment, the requesting processors could include dependency checking logic, for generating non-blocking processor requests, which may be overlapped for processing real-time transactions. A controller creates separate pending queues, preferably on a shared memory device, that correspond to each one of the peripheral devices. When requesting processors generate processor requests directed to a particular peripheral device, the controller dynamically adds entries to the corresponding pending queue, which varies its size. The controller adds the entries, which contain the processor requests, from an allocated free pool of entries that include pointers pointing to specific memory locations on the shared memory device. Therefore, at least two separate peripheral devices could process corresponding processor requests simultaneously, after retrieving such processor requests from the corresponding separate pending queues, and thereby increasing the system's transaction throughput.

According to more detailed features of the present invention, the requesting processors generate the processor requests over a first bus and the peripheral devices accept the processor requests over a second bus, which may have a different bus and width than the first bus. When the processor requests are accepted by the peripheral devices, the controller marks such processor requests as outstanding. After the peripheral devices respond to the outstanding processor requests, the controller places the processor

requests in a return queue the entries of which are freed up once corresponding responses of such requests are transmitted to the requesting processors.

According to other more detailed features of the invention, the processor requests are prioritized in the pending queues such that the higher priority processor requests are processed before the lower priority processor requests. Similarly, the separate pending queues may be prioritized such that the higher priority pending queues have a higher number of maximum entries than the lower priority pending queues.

Claim 61-70 are directed to a method for processing data according to the present invention. Similarly, claims 71-80 are directed to an apparatus for processing the data according to the present invention.

Applicants respectfully submit that none of the new claims 39-80 are anticipated by or are obvious over the Bell patent alone or when it is combined with other references. The Bell patent discloses a data processing system having one or more processors and I/O devices that perform split transactions, including read and write requests and replies, through a bus interface situated between a processor bus and an I/O bus. In this way, the processor and the I/O devices interchangeably act as requesting or responding agents for the split transactions. By deferring or committing to a request, the bus interface processes the requests and the replies in an out-of-order fashion. When a request is deferred, a requesting agent does not hold up its respective bus, to receive the reply from the responding agent. Conversely, when a request is committed to, the requesting agent holds up the bus until the reply is received from the responding agent and delivered to the bus interface.

According to the Bell patent, each one of the requesting agents maintain identical In-Order Queues disposed on the bus interface, for adding the requests and replies to such queues. The bus interface includes an outbound request queue, an inbound request queue and a transaction arbitration unit (TAU), which maintains transaction ordering between the inbound and outbound queues. An outbound request decoder decodes the requests stored in the In-Order Queues in a first-in-first-out manner, to determine whether the bus interface should commit to or defer a particular request. The requests are then placed in the outbound queue, to be processed by the responding agents. Depending on whether the request is deferred or committed to, the reply is either placed in the inbound queue or returned directly to the bus interface. When the request is deferred, the reply is placed in the inbound queue and the respective bus is not stalled. Conversely, when the request is committed to, the reply is returned to the bus interface and the respective bus is stalled. Under this arrangement, the TAU includes logic for avoiding deadlocks on the processor bus and the I/O bus, when both are stalled.

The Bell patent, however, fails to disclose one of the most important features of the present invention. That is, creating separate pending queues that correspond to each one of the peripheral devices, to allow two or more of such peripheral devices to simultaneously execute processor requests by retrieving the requests from their respective pending queues. As shown in Figure 4, the Bell patent discloses a single queue (either outbound or inbound) that is predisposed, as opposed to being dynamically created, in the bus interface. Furthermore, such single queue is shared by multiple requesting or responding agents, instead of being specific to a particular agent. Although

the Bell patent describes each agent as having corresponding In-Order queues for receiving the requests, unlike the present invention, the queues do not correspond to the responding agents. According to the Bell patent, the agents retrieve the requests from a single queue (either inbound or outbound) in a FIFO manner. The Examiner's attention is respectfully directed to column 8, lines 20-31, where the outbound request queue is described as a conventional first-in-first out (FIFO) queue having four slots containing up to four outbound transactions.

In regards to original claim 10, which covers the separate pending queues feature of the present invention, the office action states "Bell et al. teach that a plurality of queues comprise a pending queue corresponding to each of the plurality of peripheral devices connected to the I/O bus (see col. 9, lines 53-61)." Applicants respectfully submit that no teaching or suggestion could be found in the cited portion of column 9, which describes creating pending queues that correspond to each one of the plurality of peripheral devices. The cited portion simply describes how the pending requests and replies in the queues are transferred to the processor bus via the bus interface. According to the cited portion, in response to a signal that is generated when one or more requests or replies are pending in a queue, a processor state machine arbitrates access to the processor bus. Therefore, the Applicants are at a loss as to how the Examiner has concluded that pending queues corresponding to each one of the plurality of peripheral devices are taught by the cited portion.

In fact, in Applicant's view, the cited portion actually teaches away from the present invention. In column 9, lines 58-61, the Bell patent describes that "upon

being granted access, processor state machine 417 transfers the pending transaction(s) in queue 430 to the target agents on the processor bus." This statement clearly indicates that the Bell patent teaches using a single queue, i.e. queue 430, for transferring pending transactions to multiple agents. Having carefully reviewed the Bell patent in its entirety, the Applicants respectfully submit that there is no disclosure, express or implied, for creating separate queues that correspond to particular agents, such that two or more of such agents could retrieve the requests from the separate queues, to process such requests simultaneously.

The Examiner has rejected claims 12-13 and 16 under 35 U.S.C. 103 as being unpatentable over the Bell patent in view of U.S. Patent No. 5,526,508 issued to Park et al. (the Park patent). The Park patent discloses a cache controller that reduces the time delay due to write-back cycles, when a missed cache line condition is present. By buffering the write-back data in a write-back buffer and buffering the read data in a read buffer, the write-back data is written into the write-back buffer, while data is read from the main memory. The office action argues that "it would have been obvious...to have the teachings of Park et al. into teachings of Bell et al. because it would prevent the time delay to the write back buffering."

In order to make a prima facie case of obviousness based on a combination of two or more references, there has to be a suggestion in at least one of the references, to motivate one of ordinary skill in the art to try such combination. The argument set forth in the office action seems to assume that the Bell patent would benefit from improving write back cycles relating to missed cache lines. As stated above, the Bell

patent improves transaction throughput between a processor bus and an I/O bus by using the inbound and the outbound buffers. Accordingly, write back cycle improvement of the Park patent, which relate to handling of missed cache lines by the cache controller, seems unrelated to increasing the data throughput between processors and I/O devices.

Even assuming arguendo that the problems solved by these references are related, the teachings of the Park patent do not add to the teachings of the Bell patent in any way. Both references use two buffers, to buffer data during read and write transactions. None of the references, however, teach or suggest creating separate queues from which a corresponding number of peripheral devices could retrieve transaction requests, to simultaneously process such requests.

Finally, the Examiner has rejected claims 21-26 under 35 U.S.C. 103 as being unpatentable over the Bell patent in view of U.S. Patent No. 5,542,055 issued to Amini et al. (the Amini patent). The Amini patent discloses a system that counts the number of peripheral devices connected to a hierarchial bus configuration, such as the PCI bus configuration, and creates a map of the peripheral buses within the hierarchial bus configuration, to locate the peripheral devices. Similar to the other two cited references, the Applicants have been unable to find any teaching or suggestion in the Amini patent that relates to the use of separate and independent queues that correspond to the peripheral devices. Accordingly, the Amini patent cannot be relied on alone or in combination with the Bell patent to reject the new claims of the present invention.

In view of the above amendments and arguments, the Applicants believe that the new claims 39-80 are in condition for allowance and solicit the Examiner to issue an early notice of allowance.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

By: _____
Robert S. Babayi
Registration No. 33,471

Post Office Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620

Date: December 20, 1996

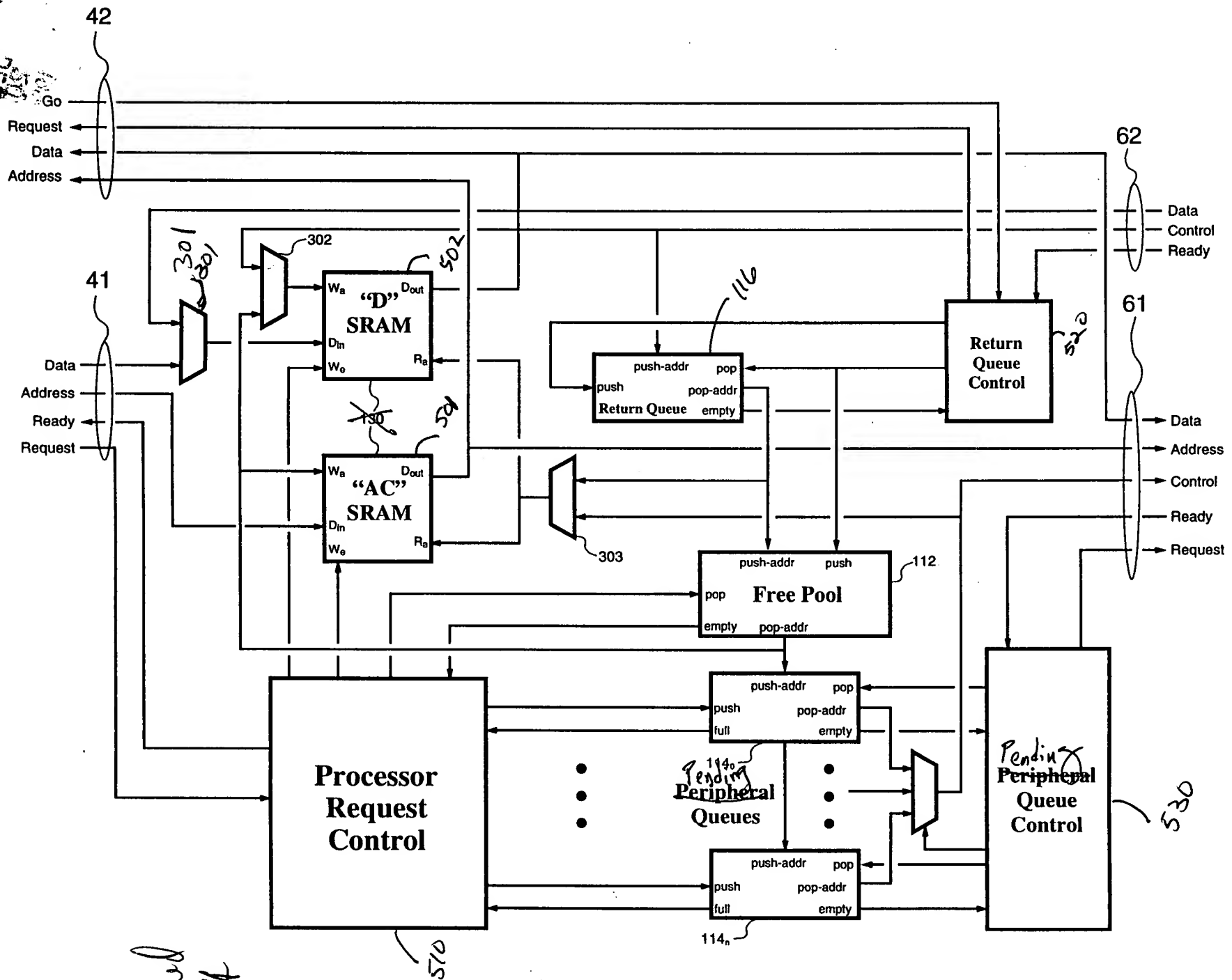
42
Go
Request
Data
Address
62
Data
Control
Ready
41
Data
Address
Ready
Request
301
302
"D" SRAM
Wa
Dout
Din
We
Ra
502
130
504
"AC" SRAM
303
116
push-addr
pop
push
pop-addr
Return Queue
empty
Return Queue Control
512
61
Data
Address
Control
Ready
Request
112
Free Pool
push-addr
push
pop
empty
pop-addr
full
Processor Request Control
510
114₀
Pending Peripheral Queues
114n
Pending Peripheral Queue Control
530

Figure 10